Advances in Engineering Software 37 (2006) 432–438
www.elsevier.com/locate/advengsoft

An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator

Ali T. Hasan, A.M.S. Hamouda *, N. Ismail, H.M.A.A. Al-Assadi

Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, University Putra Malaysia, 43400UPM Serdang, Selangor, Malaysia

Received 19 November 2004; received in revised form 4 August 2005; accepted 30 September 2005
Available online 3 February 2006

Abstract

An adoptive learning strategy using an artificial neural network ANN has been proposed here to control the motion of a 6 D.O.F manipulator robot and to overcome the inverse kinematics problem, which are mainly singularities and uncertainties in arm configurations. In this approach a network have been trained to learn a desired set of joint angles positions from a given set of end effector positions, experimental results has shown an excellent mapping over the working area of the robot, to validate the ability of the designed network to make prediction and well generalization for any set of data, a new training using different data set has been performed using the same network, experimental results has shown a good generalization for the new data sets.

The proposed control technique does not require any prior knowledge of the kinematics model of the system being controlled, the basic idea of this concept is the use of the ANN to learn the characteristics of the robot system rather than to specify explicit robot system model. Any modification in the physical set-up of the robot such as the addition of a new tool would only require training for a new path without the need for any major system software modification, which is a significant advantage of using neural network technology.

Keywords: Neural networks; Inverse kinematics; Back propagation; Robot control

1. Introduction

Trajectory control of robotic manipulators traditionally consists of following a pre-programmed sequence of end effector movements. Robot control usually requires control signals applied at the joints of the robot while the desired trajectory, or sequence of arm end positions, is specified for the end effector. The robot arms operate in a plane. To make the arm move, desired coordinates of the end effector point are fed to the robot controller for generating the joint angles for the motors that move the arms. To perform end effector position control of a robotic manipulator, the inverse kinematics problem (IK), needs to be solved [1].

Finding a solution to the IK problem analytically for serial manipulators is a difficult problem and a focus of much researches [2–12]. One source of difficulty is due to the fact that the inverse is not a true function; the set of solutions is infinite. In addition, the inverse map-ping is non-linear, and the nature of the inverse solution set for particular target locations undergoes a qualitative change as the manipulator moves between different regions of the workspace.

If these solutions provided were reasonably efficient in solving the problem, they have several drawbacks also, Firstly the complexity of the model requires high computational time and if the specific model-structure does not properly reflect all the robot characteristics, it can result in a poor control performance, Secondly the fact that the model is highly system-specific makes it very hard to accommodate physical changes such as the addition of a new tool [13].

The overall complexity of robot control problem and the quest for a truly autonomous robot system has led to considerable interest being devoted to the application of neural network technology to robot control [14–18]. ANNs are widely accepted as a technology offering an alternative way to tackle complex and ill-defined problems. They can learn from examples, are fault tolerant in the sense that they are able to handle noisy and incomplete data, are able to deal with non-linear problems and, once trained, can perform prediction and generalization at high speed, They are particularly useful in system modelling such as implementing complex mappings [19].

To overcome the uncertainties and non-linearity of the robot model, an ANN have been designed to learn the characteristics

* Corresponding author. Tel.: +603 89466330; fax: +603 89422107.
E-mail address: hamouda@eng.upm.edu.my (A.M.S. Hamouda).

0965-9978/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2005.09.010

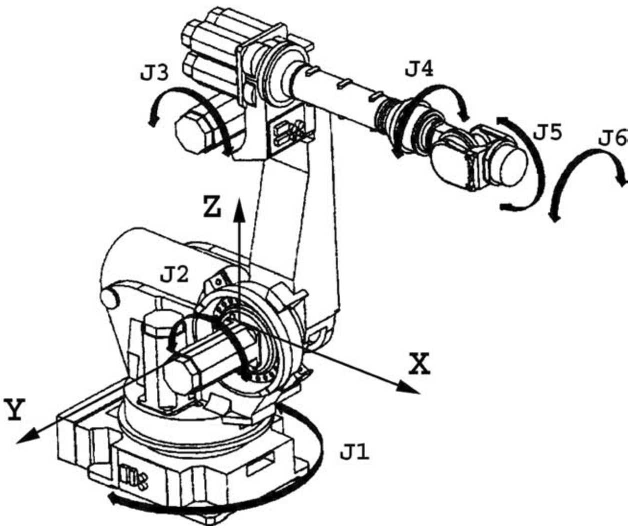

Fig. 1. FANUC M710i Robot, Main axes, wriest axes and global coordinate system.

of a FANUC M-710i robot over particular region of the working space, the FANUC M-710i as can be seen in Fig. 1, is a serial robot manipulator consisting of axes and arms driven by servomotors. The place at which arm is connected is a joint, or an axis. J1, J2, and J3 are main axes. The basic configuration of the robot depends on whether each main axis functions as a linear axis or rotation axis.

The wrist axes are used to move an end effecter (tool) mounted on the wrist flange. The wrist itself can be wagged about one wrist axis and the end effecter rotated about the other wrist axis, this highly non-linear structure makes this robot very useful in typical industrial applications such as the material handling, assembly of parts, painting, etc.

This paper is devoted to the develop of a new adaptive ANN controller to track IK control problem of a 6 D.O.F serial robot manipulator, the learning algorithm is based on the adaptive updating of the weights of the network by minimizing the tracking error after each iteration process.

2. Robot arm kinematics configuration

The inverse kinematics problem is the problem of finding a vector of joint variables that produce a desired end effector location. If a unique vector of joint angles exists which attains the desired end effector location, there is a well-defined inverse to the forward kinematics function and the inverse kinematics problem is well posed. Unfortunately, the inverse kinematics problem can be ill-posed because the solution for the forward kinematics is not unique; in many cases solving the inverse kinematics problem may result in infinite number of solutions [20].

The kinematics considerations for the manipulator shown in Fig. 2 are based on the forward kinematics equation. The forward kinematics equation involves mapping of joint angle coordinates (θ_1, θ_2) to the end effector position (x, y). The mapping expressions can be obtained by inspection of the figure as follows:

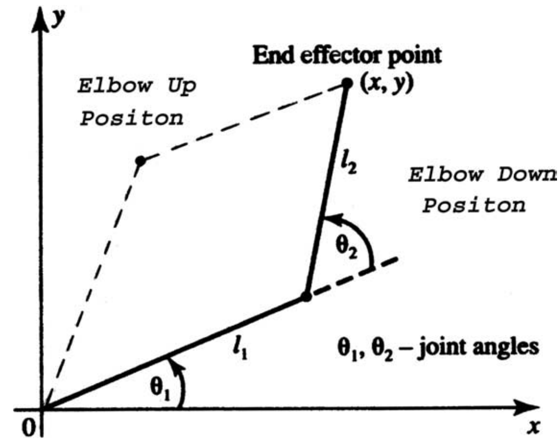

Fig. 2. The two arm configurations positioning at (x, y).

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (1)$$

$$y = l_2 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

Where θ_1 and θ_2 are the joint angles of the first and second arm segments, respectively, l_1, and l_2 are respective arm segment lengths. Relation (1) expresses the forward kinematics problem and implements unique mapping from the joint angle space to the Cartesian space.

The inverse kinematics problem is described as follows:

$$\theta_2 = \cos^{-1}\left[\frac{(x^2 + y^2 - l_1^2 - l_2^2)}{(2 l_1 l_2)}\right] \quad (2a)$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left[\frac{l_2 \sin \theta_2}{(l_1 + l_2 \cos \theta_2)}\right] \quad (2b)$$

Since ($\cos^{-1}$) is not a single-valued function in the range of angles of interest, two possible orientations typically result from relation (2) for the robot arm joint angles. The arm can be positioned with the elbow up or down, with the end effector still at the required (x, y) point. The inverse kinematics transformation (2) implementing mapping from Cartesian space to joint space is thus not unique [21].

3. The neural network strategy

The possibility of developing a machine that would 'think' has intrigued human beings since ancient times, machinery can outperform humans physically. Similarly, computers can outperform mental functions in limited areas, notably in the speed of mathematical calculations. For example, the fastest computers developed are able to perform roughly 10 billion calculations per second. But making more powerful computers will probably not be the way to create a machine capable of thinking. Computer programs operate according to set procedures, or logic steps, called algorithms. In addition, most computers do serial processing such as operations of recognition and computations are performed one at a time. The brain works in a manner called parallel processing,